```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
# Load the train data in a dataframe
train = pd.read_csv(r'C:\Users\Nithishma\Desktop\train.csv')
test = pd.read_csv(r'C:\Users\Nithishma\Desktop\test.csv')
train.info()
nulls = train.isnull().sum().sort_values(ascending=False)
nulls.head(20)
train = train.drop(['Id','PoolQC','MiscFeature','Alley','Fence'],axis = 1)
train[['Fireplaces','FireplaceQu']].head(10)
train['FireplaceQu'].isnull().sum()
train['Fireplaces'].value_counts()
train['FireplaceQu']=train['FireplaceQu'].fillna('NF')
train['LotFrontage'] =train['LotFrontage'].fillna(value=train['LotFrontage'].mean())
train['GarageType'].isnull().sum()
train['GarageCond'].isnull().sum()
train['GarageFinish'].isnull().sum()
train['GarageYrBlt'].isnull().sum()
train['GarageQual'].isnull().sum()
train['GarageArea'].value_counts().head()
train['GarageType']=train['GarageType'].fillna('NG')
train['GarageCond']=train['GarageCond'].fillna('NG')
train['GarageFinish']=train['GarageFinish'].fillna('NG')
train['GarageYrBlt']=train['GarageYrBlt'].fillna('NG')
```

```
train['GarageQual']=train['GarageQual'].fillna('NG')

train.BsmtExposure.isnull().sum()

train.BsmtFinType2.isnull().sum()

train.BsmtFinType1.isnull().sum()

train.BsmtCond.isnull().sum()

train.BsmtQual.isnull().sum()

train.TotalBsmtSF.value_counts().head()

train.TotalBsmtSF.value_counts().head()

train['BsmtExposure']=train['BsmtExposure'].fillna('NB')

train['BsmtFinType2']=train['BsmtFinType2'].fillna('NB')

train['BsmtFinType1']=train['BsmtFinType1'].fillna('NB')

train['BsmtCond']=train['BsmtCond'].fillna('NB')

train['BsmtQual']=train['BsmtQual'].fillna('NB')

train['MasVnrArea'] = train['MasVnrArea'].fillna(train['MasVnrArea'].mean())

train['MasVnrType'] = train['MasVnrType'].fillna('none')

train.Electrical = train.Electrical.fillna('SBrkr')

train.isnull().sum().sum()

num_train = train._get_numeric_data()

num_train.columns

def var_summary(x):

    return pd.Series([x.count(), x.isnull().sum(), x.sum(), x.mean(), x.median(),  x.std(), x.var(), x.min(),
x.quantile(0.01), x.quantile(0.05),x.quantile(0.10),x.quantile(0.25),x.quantile(0.50),x.quantile(0.75),
x.quantile(0.90),x.quantile(0.95), x.quantile(0.99),x.max()],

index=['N', 'NMISS', 'SUM', 'MEAN','MEDIAN', 'STD', 'VAR', 'MIN', 'P1' ,
'P5' ,'P10' ,'P25' ,'P50' ,'P75' ,'P90' ,'P95' ,'P99' ,'MAX'])

num_train.apply(lambda x: var_summary(x)).T

sns.boxplot([num_train.LotFrontage])

train['LotFrontage']= train['LotFrontage'].clip(upper=train['LotFrontage'].quantile(0.99))

sns.boxplot(num_train.LotArea)
```

```
train['LotArea']= train['LotArea'].clip(upper=train['LotArea'].quantile(0.99))
sns.boxplot(train['MasVnrArea'])
train['MasVnrArea']= train['MasVnrArea'].clip(upper=train['MasVnrArea'].quantile(0.99))
sns.boxplot(train['BsmtFinSF1'])
sns.boxplot(train['BsmtFinSF2'])
train['BsmtFinSF1']= train['BsmtFinSF1'].clip(upper=train['BsmtFinSF1'].quantile(0.99))
train['BsmtFinSF2']= train['BsmtFinSF2'].clip(upper=train['BsmtFinSF2'].quantile(0.99))
sns.boxplot(train['TotalBsmtSF'])
train['TotalBsmtSF']= train['TotalBsmtSF'].clip(upper=train['TotalBsmtSF'].quantile(0.99))
sns.boxplot(train['1stFlrSF'])
train['1stFlrSF']= train['1stFlrSF'].clip(upper=train['1stFlrSF'].quantile(0.99))
sns.boxplot(train['2ndFlrSF'])
train['2ndFlrSF']= train['2ndFlrSF'].clip(upper=train['2ndFlrSF'].quantile(0.99))
sns.boxplot(train['GrLivArea'])
train['GrLivArea']= train['GrLivArea'].clip(upper=train['GrLivArea'].quantile(0.99))
sns.boxplot(train['BedroomAbvGr'])
train['BedroomAbvGr']= train['BedroomAbvGr'].clip(upper=train['BedroomAbvGr'].quantile(0.99))
train['BedroomAbvGr']= train['BedroomAbvGr'].clip(lower=train['BedroomAbvGr'].quantile(0.01))
sns.boxplot(train['GarageCars'])
train['GarageCars']= train['GarageCars'].clip(upper=train['GarageCars'].quantile(0.99))
sns.boxplot(train['GarageArea'])
train['GarageArea']= train['GarageArea'].clip(upper=train['GarageArea'].quantile(0.99))
sns.boxplot(train['WoodDeckSF'])
train['WoodDeckSF']= train['WoodDeckSF'].clip(upper=train['WoodDeckSF'].quantile(0.99))
sns.boxplot(train['OpenPorchSF'])
train['OpenPorchSF']= train['OpenPorchSF'].clip(upper=train['OpenPorchSF'].quantile(0.99))
sns.boxplot(train['EnclosedPorch'])
```

```
train['EnclosedPorch']= train['EnclosedPorch'].clip(upper=train['EnclosedPorch'].quantile(0.99))
sns.boxplot(train['3SsnPorch'])
train['3SsnPorch']= train['3SsnPorch'].clip(upper=train['3SsnPorch'].quantile(0.99))
sns.boxplot(train['ScreenPorch'])
train['ScreenPorch']= train['ScreenPorch'].clip(upper=train['ScreenPorch'].quantile(0.99))
sns.boxplot(train['PoolArea'])
train['PoolArea']= train['PoolArea'].clip(upper=train['PoolArea'].quantile(0.99))
sns.boxplot(train['MiscVal'])
sns.boxplot(train.SalePrice)
train['SalePrice']= train['SalePrice'].clip(upper=train['SalePrice'].quantile(0.99))
train['SalePrice']= train['SalePrice'].clip(lower=train['SalePrice'].quantile(0.01))
train['MiscVal']= train['MiscVal'].clip(upper=train['MiscVal'].quantile(0.99))
num_corr=num_train .corr()
plt.subplots(figsize=(13,10))
sns.heatmap(num_corr,vmax =.8 ,square = True)
k = 14
cols = num_corr.nlargest(k, 'SalePrice')['SalePrice'].index
cm = np.corrcoef(num_train[cols].values.T)
sns.set(font_scale=1.35)
f, ax = plt.subplots(figsize=(10,10))
hm=sns.heatmap(cm, annot = True,vmax =.8, yticklabels=cols.values, xticklabels = cols.values)
from sklearn.preprocessing import StandardScaler
train_d = pd.get_dummies(train)
train_d1 = train_d.drop(['SalePrice'],axis = 1)
y = train_d.SalePrice
scaler = StandardScaler()
scaler.fit(train_d1)
```

```
t_train = scaler.transform(train_d1)

from sklearn.decomposition import PCA

pca_hp = PCA(30)

x_fit = pca_hp.fit_transform(t_train)

np.exp(pca_hp.explained_variance_ratio_)
```